PROFET: No-stress Performance Prediction

Mariana Carmin

Barcelona Supercomputing Center

25th of February 2026.

It's the Memory, Stupid!

Outline

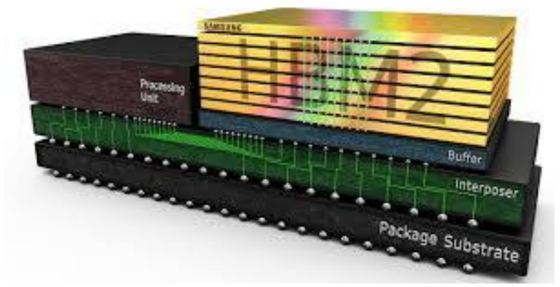
- How everything started
- General idea
- General workflow
  - System profiling
  - Application profiling
- Putting it all together: System & Application profiling
- Brutal evaluation [!]
- Demo / Hands on
- Explore more

How everything started

- The whole story started with in 2014
  - Project objective: Memory design space exploration
    - Target CPU: High-end Intel (main-stream)
    - Main memory: Different options

How everything started

- The whole story started with in 2014
  - Project objective: Memory design space exploration
    - Target CPU: High-end Intel (main-stream)
    - Main memory: Different options



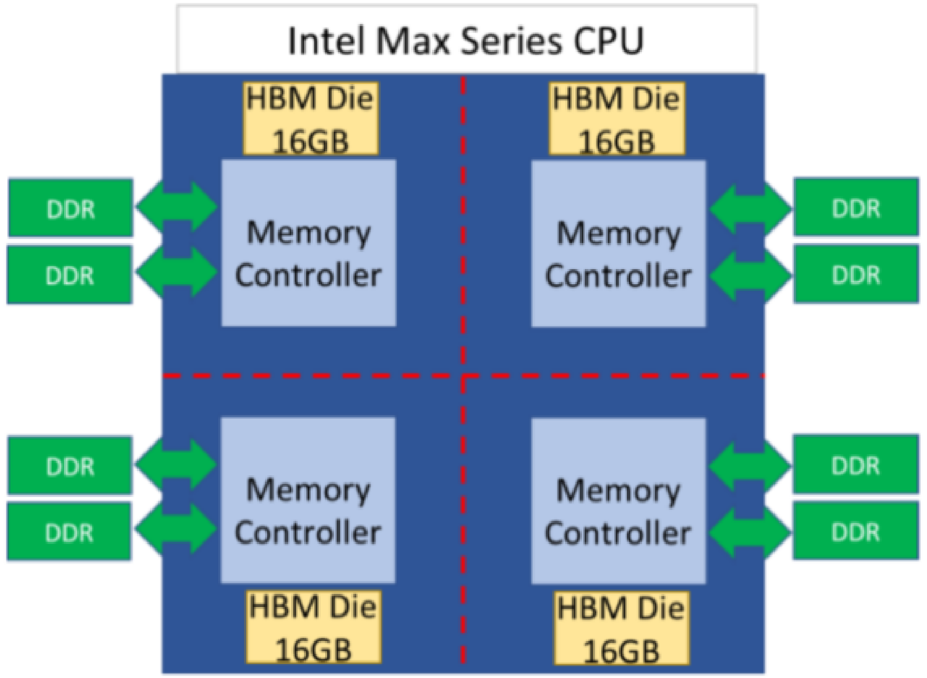
CXL Compute Express Link®

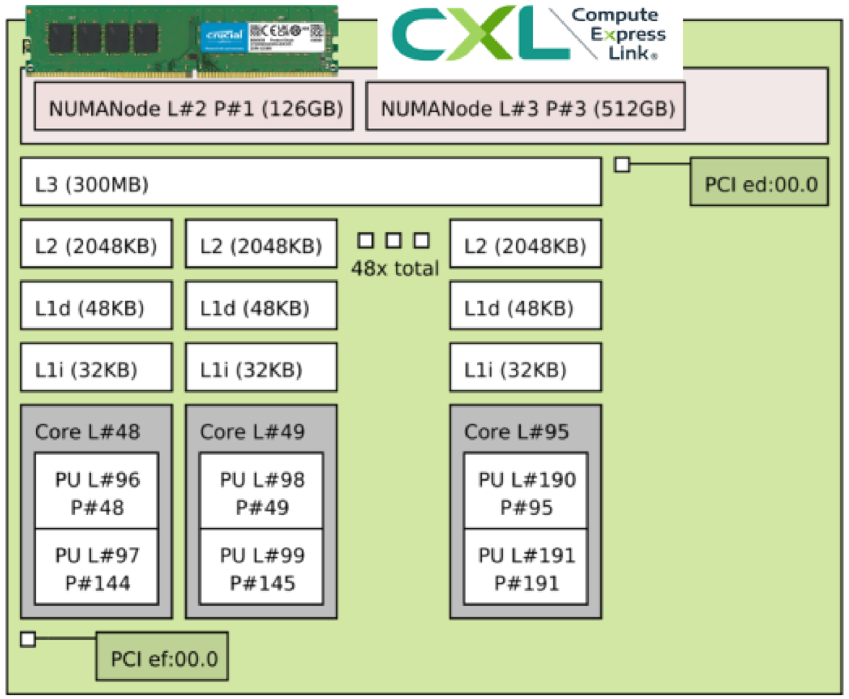
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

It's the Memory, Stupid!

How everything started

- The whole story started with in 2014
  - Project objective: Memory design space exploration
    - Target CPU: High-end Intel (main-stream)
    - Main memory: Different options





How everything started

- The whole story started with in 2014
  - Project objective: Memory design space exploration
    - Target CPU: High-end Intel (main-stream)
    - Main memory: Different options

- Conventional approach: Detailed hardware simulation (Zsim + DRAMSim2)

How everything started

- The whole story started with in 2014
  - Project objective: Memory design space exploration
    - Target CPU: High-end Intel (main-stream)
    - Main memory: Different options

- Conventional approach: Detailed hardware simulation (Zsim + DRAMSim2)
  - Frustration
    - Pain, agony and torture inherent to detailed hardware simulation
    - Time consuming
    - We had to simulate an existing CPU **only** to drive our memory system simulation

  - Idea: ***There has to be a better way to do this!***

PROFET: Memory system design space exploration

- Answering the question:
  "How much will performance, power and energy of my application change if we change the main memory?"

- We profile an application on a **baseline system**
  - Baseline system is an **existing server** (with existing memory system), e.g. Intel GNR with RDIMM-6400

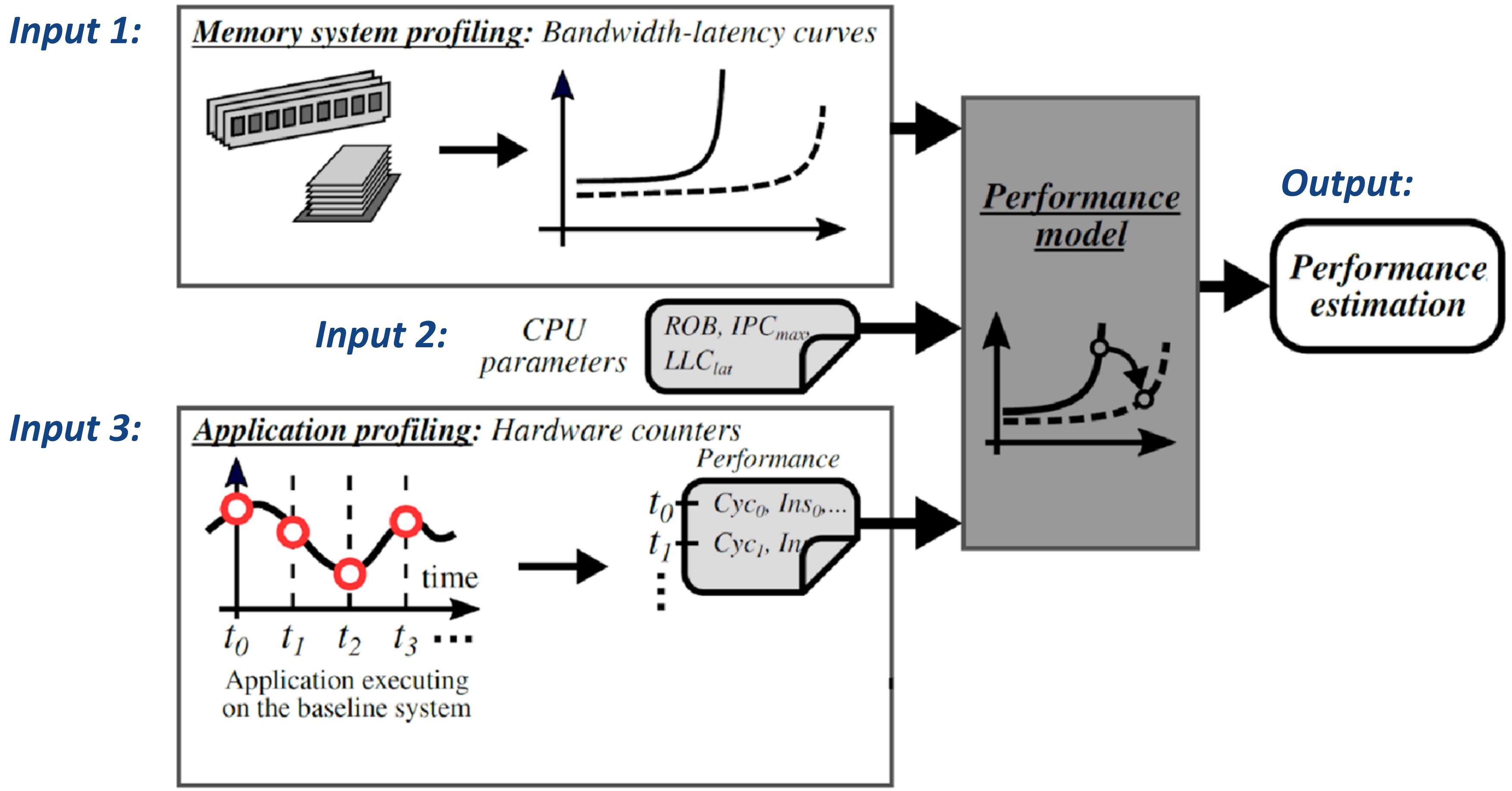
- **Target system**
  - The **CPU remains the same**, but we want to explore the performance, power and energy **implications of a new memory system on my application**:
    - E.g. How much would my application benefit from the RDIMM-6400 → MRDIMM-8800 upgrade of the server

Published in 2019 – Still relevant

- First big tool developed by the BSC memory team
- Used by various industrial partners: **Samsung, Micron, Huawei**
- PROFET passed the test of time
  - Tool/method developed for SandyBridge with DDR3
  - Plug&Play
    - Intel Knights Landing (DDR4, MCDRAM), Cascade Lake (DDR4), Sapphire Rapids (DDR5), Emerald Rapids (DDR5, CXL memory expanders), Granite Rapids (DDR5 RDIMMs and MRDIMMs)
    - Huawei Kunpeng 920 (DDR4)
  - Led to the development of the Mess framework

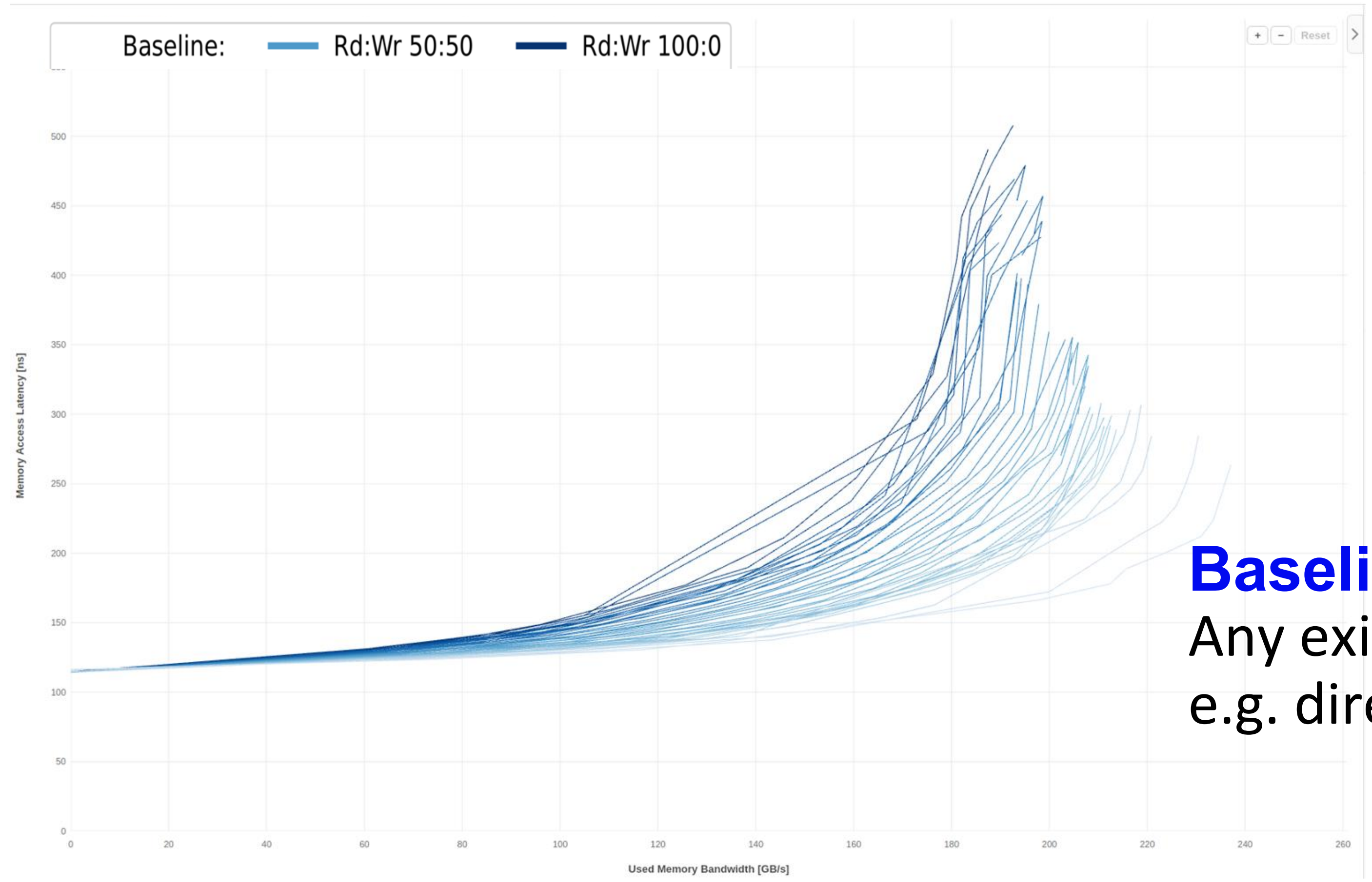
Milan Radulovic, Rommel Sánchez Verdejo, Paul Carpenter, Petar Radojković, Bruce Jacob, and Eduard Ayguadé. ***PROFET: Modeling System Performance and Energy Without Simulating the CPU***. SIGMETRICS 2019.

Overall workflow


Input 1:
Memory system profiling: Bandwidth-latency curves
Input 2:
CPU parameters
ROB, IPC$_{max}$, LLC$_{lat}$
Input 3:
Application profiling: Hardware counters
time
t_0 t_1 t_2 t_3 ...
Application executing on the baseline system
Performance
t_0 Cyc$_0$, Ins$_0$,...
t_1 Cyc$_1$, In
Performance model
Output:
Performance estimation

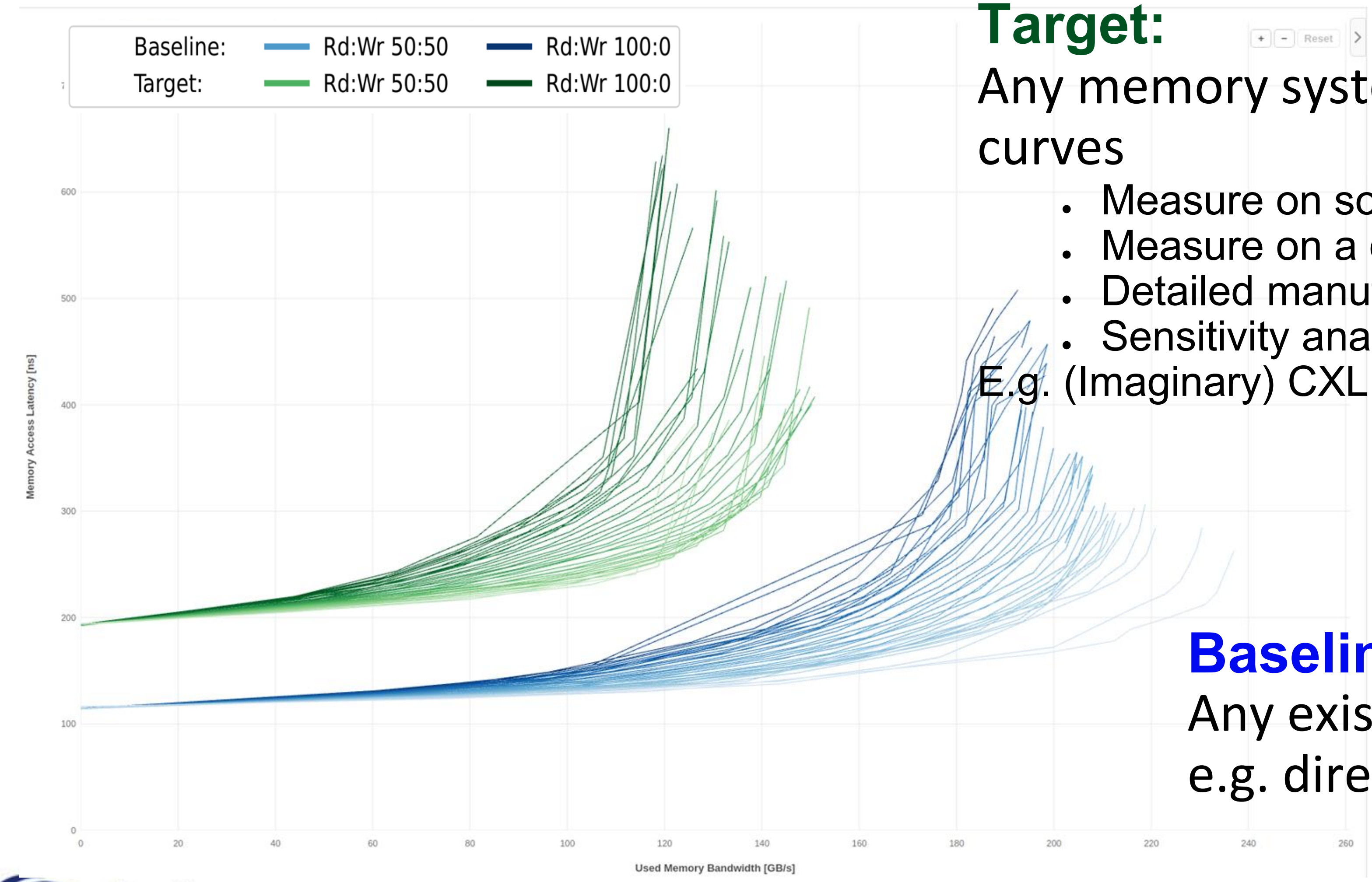
Note: In this presentation we focus on the performance estimate. For power and energy estimate, please refer to the PROFET paper (SIGMETRICS 2019).

Input 1:
Memory system profiling: Bandwidth–latency curves



Baseline:
Any existing memory system
e.g. directly attached DDR4/5

Input 1:
Memory system profiling: Bandwidth–latency curves



Target:

Any memory system for which you can get the curves

- Measure on some actual system
- Measure on a dev-board or prototype
- Detailed manufacturers simulation/estimate
- Sensitivity analysis

E.g. (Imaginary) CXL memory expander

Baseline:

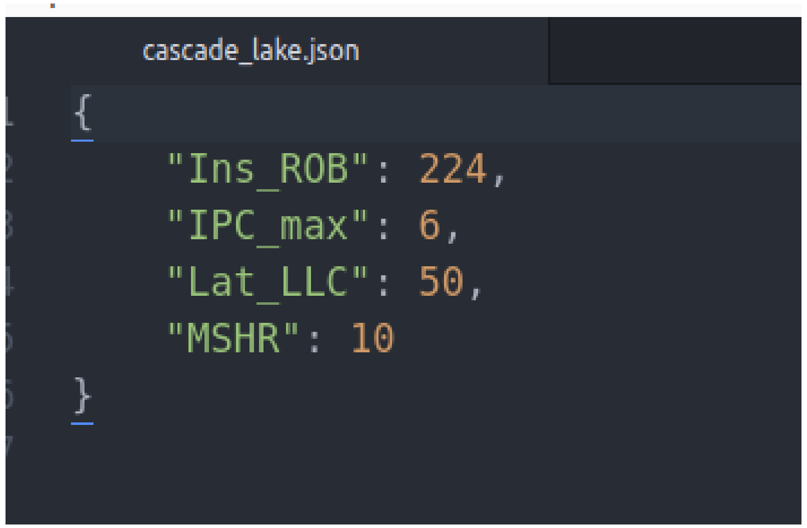
Any existing memory system
e.g. directly attached DDR4/5

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

It's the Memory, Stupid!

Input 2: CPU parameters

- Some (basic) CPU parameters: Typically available in the CPU datasheets
  - Re-order buffer (ROB) capacity
  - Miss information status holding register(MSHR) capacity
  - Minimum theoretical cycles-per-instruction (CPI)
  - Latency of last level cache (LLC)


cascade_lake.json

```
{
    "Ins_ROB": 224,
    "IPC_max": 6,
    "Lat_LLC": 50,
    "MSHR": 10
}
```

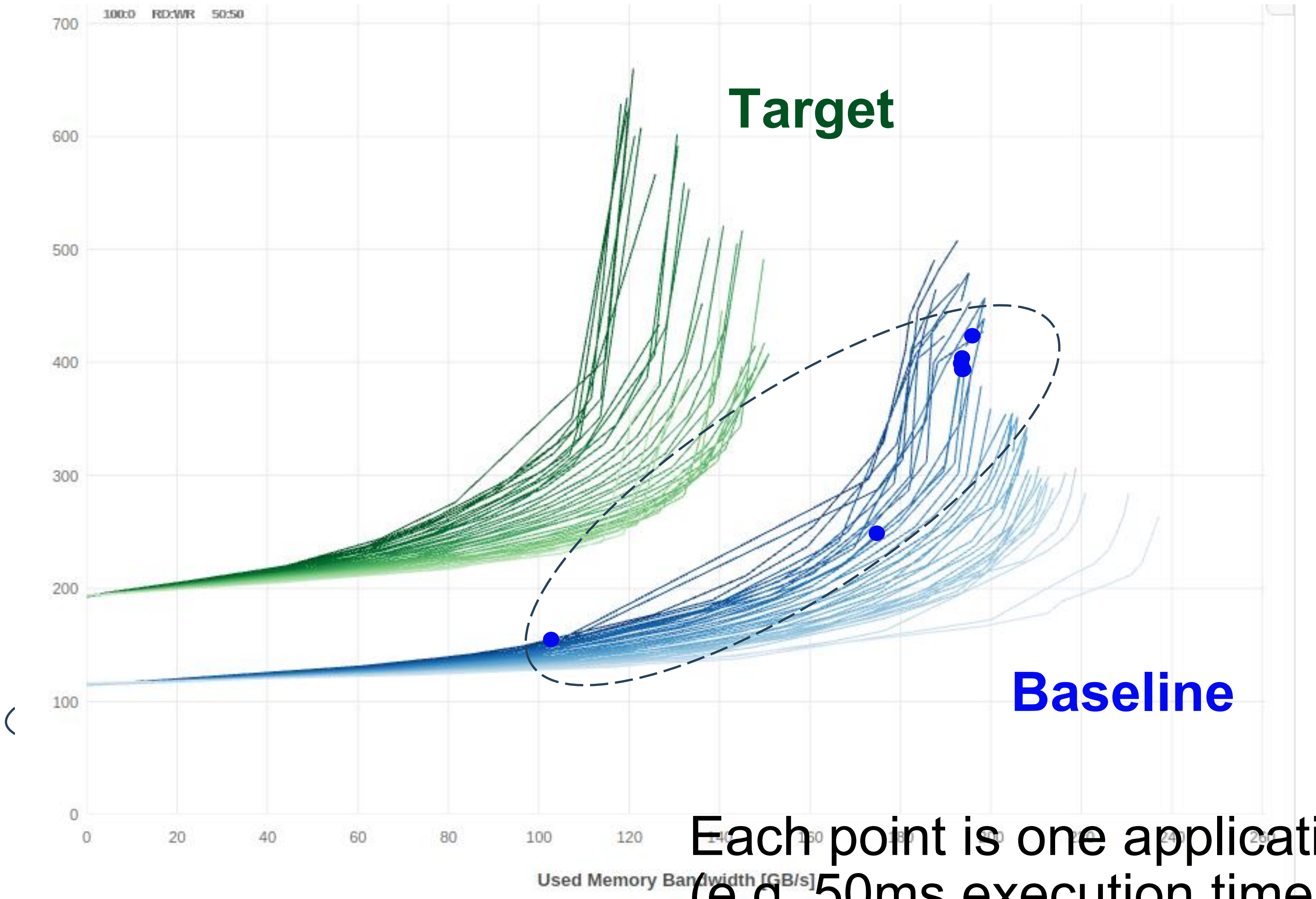
Input 3: Application profiling

- Done on the baseline (actual) memory system, e.g. Intel GRP with RDIMM-6400

- Execute application of the system and read hardware counters
  - Read and write memory bandwidths → The same as the Mess application profiling
  - CPU cycles
  - Number of instructions
  - Number of LLC misses

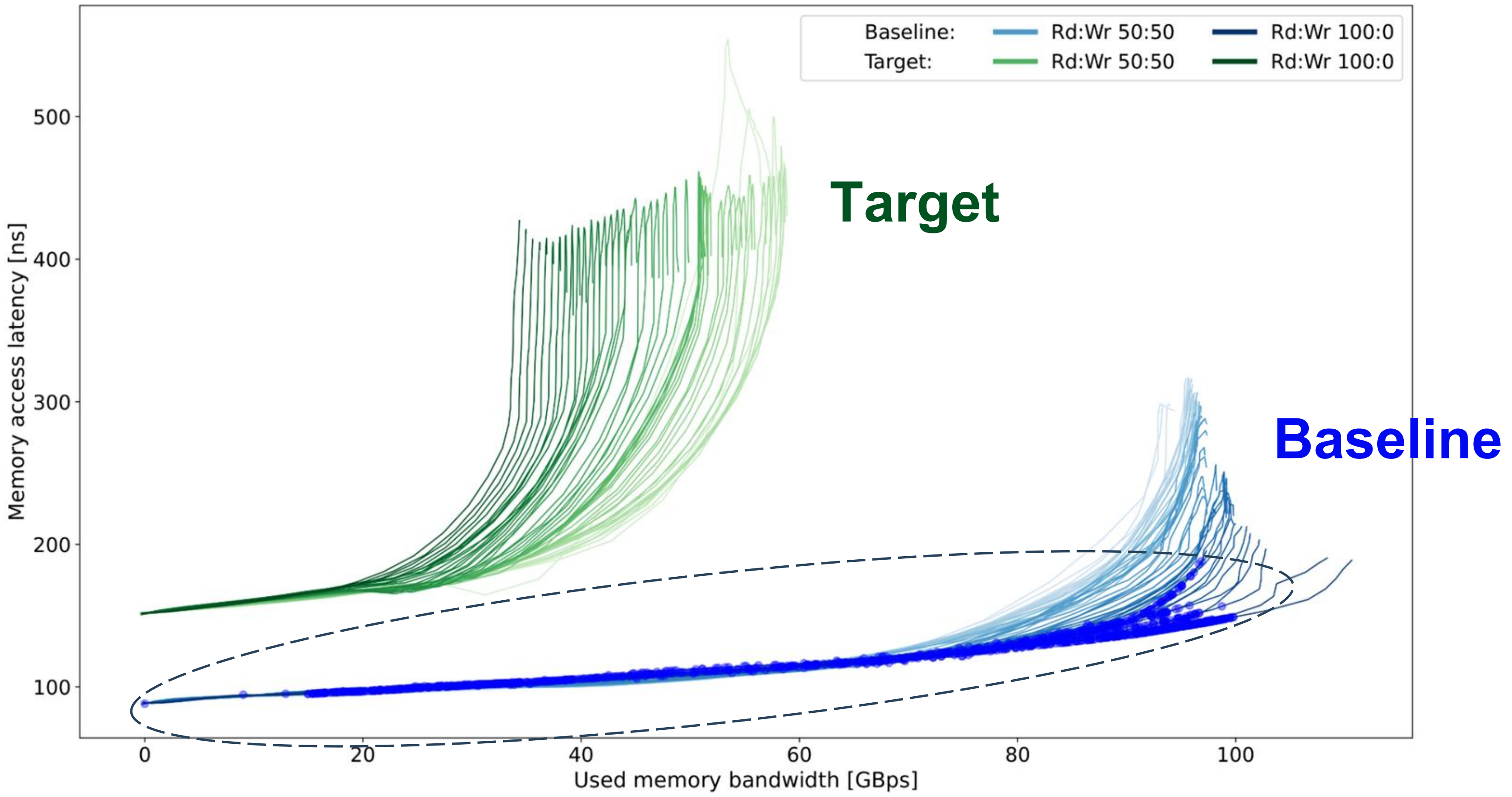
- Sampling in time
  - Reasonable sampling interval: >10ms

Analysis Step 1: Position your app on the curves



Each point is one application observation (e.g. 50ms execution time)

Analysis Step 1: Position your app on the curves



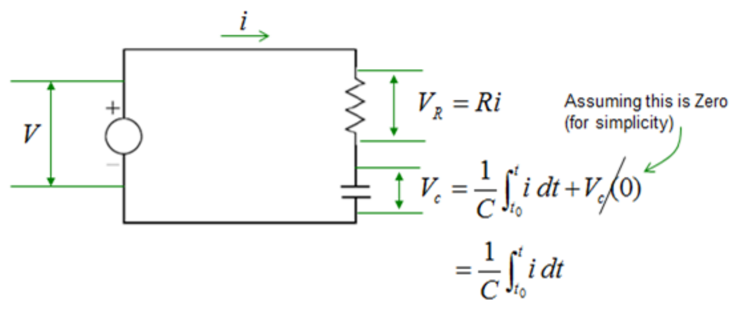
Each point is one application observation (e.g. 10ms execution time)

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

It's the Memory, Stupid!

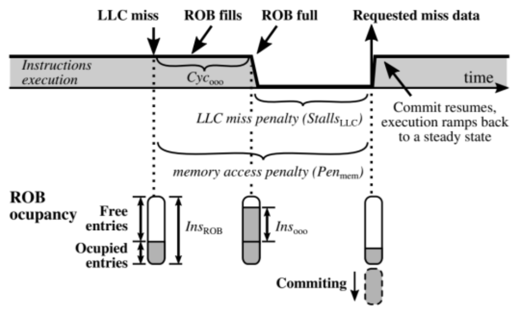
PROFET performance model

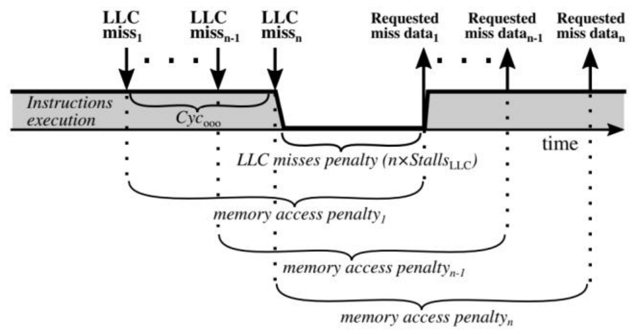
- **Old-school electrical engineering mindset**
  - Model a circuit with a system of equations → Solve the system



i
V
$V_R = Ri$
Assuming this is Zero (for simplicity)
$V_c = \frac{1}{C}\int_{t_0}^{t} i\,dt + \cancel{V_c(0)}$
$= \frac{1}{C}\int_{t_0}^{t} i\,dt$

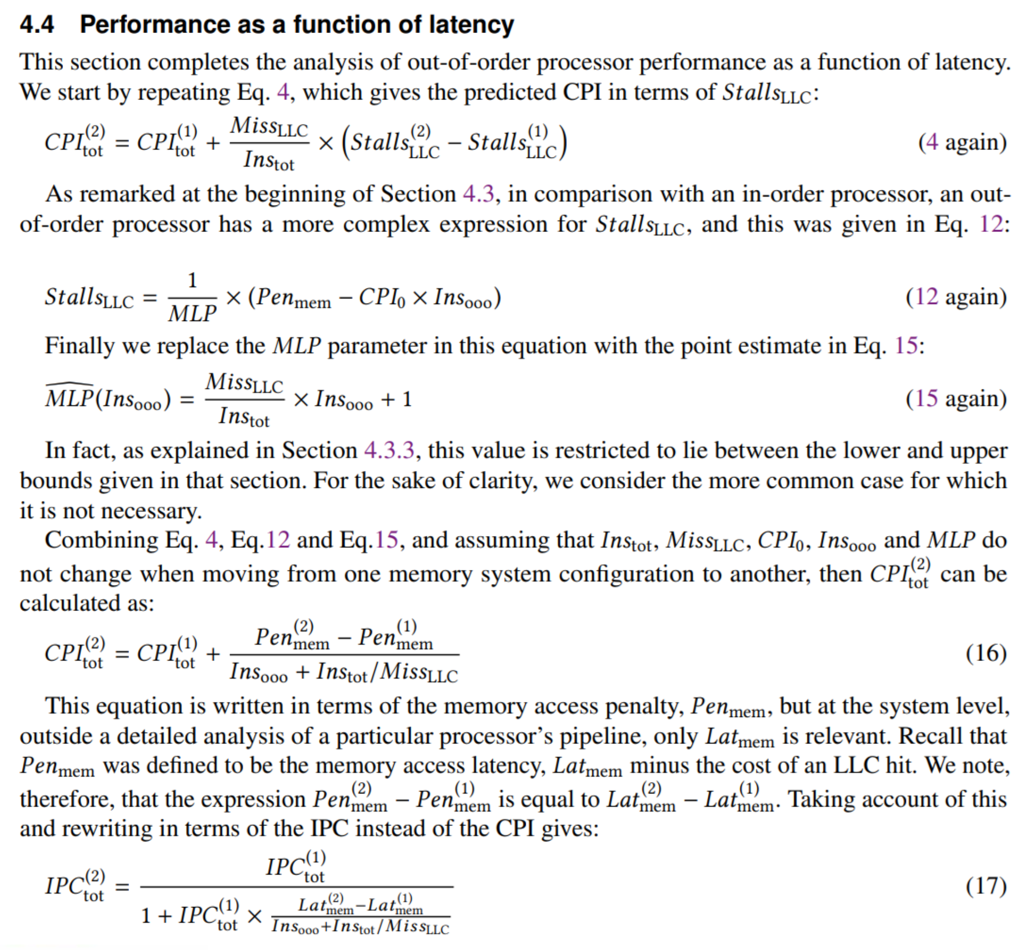
- **Unconventional approach** to memory design-space exploration





LLC miss$_1$
LLC miss$_{n-1}$
LLC miss$_n$
Requested miss data$_1$
Requested miss data$_{n-1}$
Requested miss data$_n$
Instructions execution
Cyc_{ooo}
time
LLC misses penalty ($n \times Stalls_{LLC}$)
memory access penalty$_1$
memory access penalty$_{n-1}$
memory access penalty$_n$

It's the Memor



4.4 Performance as a function of latency

This section completes the analysis of out-of-order processor performance as a function of latency. We start by repeating Eq. 4, which gives the predicted CPI in terms of $Stalls_{LLC}$:

$$CPI^{(2)}_{tot} = CPI^{(1)}_{tot} + \frac{Miss_{LLC}}{Ins_{tot}} \times \left(Stalls^{(2)}_{LLC} - Stalls^{(1)}_{LLC}\right) \qquad \text{(4 again)}$$

As remarked at the beginning of Section 4.3, in comparison with an in-order processor, an out-of-order processor has a more complex expression for $Stalls_{LLC}$, and this was given in Eq. 12:

$$Stalls_{LLC} = \frac{1}{MLP} \times (Pen_{mem} - CPI_0 \times Ins_{ooo}) \qquad \text{(12 again)}$$

Finally we replace the MLP parameter in this equation with the point estimate in Eq. 15:

$$\widehat{MLP}(Ins_{ooo}) = \frac{Miss_{LLC}}{Ins_{tot}} \times Ins_{ooo} + 1 \qquad \text{(15 again)}$$

In fact, as explained in Section 4.3.3, this value is restricted to lie between the lower and upper bounds given in that section. For the sake of clarity, we consider the more common case for which it is not necessary.

Combining Eq. 4, Eq.12 and Eq.15, and assuming that Ins_{tot}, $Miss_{LLC}$, CPI_0, Ins_{ooo} and MLP do not change when moving from one memory system configuration to another, then $CPI^{(2)}_{tot}$ can be calculated as:

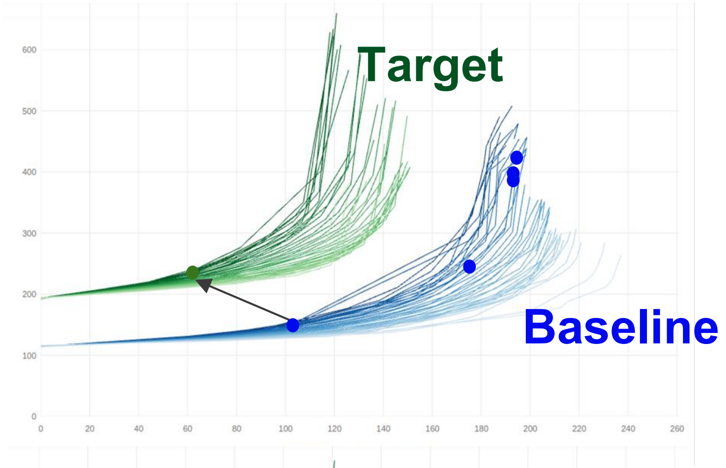
$$CPI^{(2)}_{tot} = CPI^{(1)}_{tot} + \frac{Pen^{(2)}_{mem} - Pen^{(1)}_{mem}}{Ins_{ooo} + Ins_{tot}/Miss_{LLC}} \qquad (16)$$

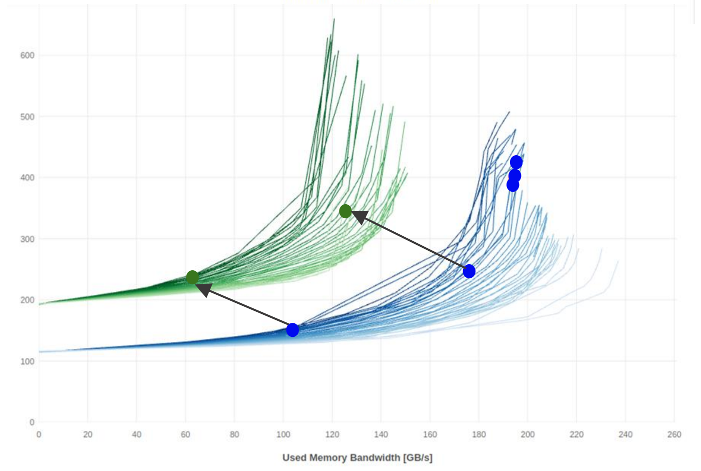
This equation is written in terms of the memory access penalty, Pen_{mem}, but at the system level, outside a detailed analysis of a particular processor's pipeline, only Lat_{mem} is relevant. Recall that Pen_{mem} was defined to be the memory access latency, Lat_{mem} minus the cost of an LLC hit. We note, therefore, that the expression $Pen^{(2)}_{mem} - Pen^{(1)}_{mem}$ is equal to $Lat^{(2)}_{mem} - Lat^{(1)}_{mem}$. Taking account of this and rewriting in terms of the IPC instead of the CPI gives:

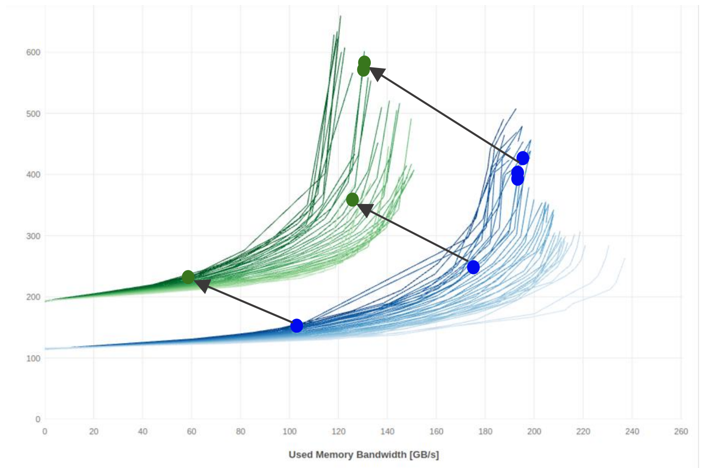
$$IPC^{(2)}_{tot} = \frac{IPC^{(1)}_{tot}}{1 + IPC^{(1)}_{tot} \times \frac{Lat^{(2)}_{mem} - Lat^{(1)}_{mem}}{Ins_{ooo} + Ins_{tot}/Miss_{LLC}}} \qquad (17)$$

PROFET predicts how application "moves" between the curves

- **Idea 1:**
  - Application can be positioned on the memory curves
  - I hope that by now we agree on this
- **Idea 2:**
  - We can predict (calculate) the application's position on the target curves
    - Full system of equations → Paper & Git
    - In this presentation: Trust me that this can be done

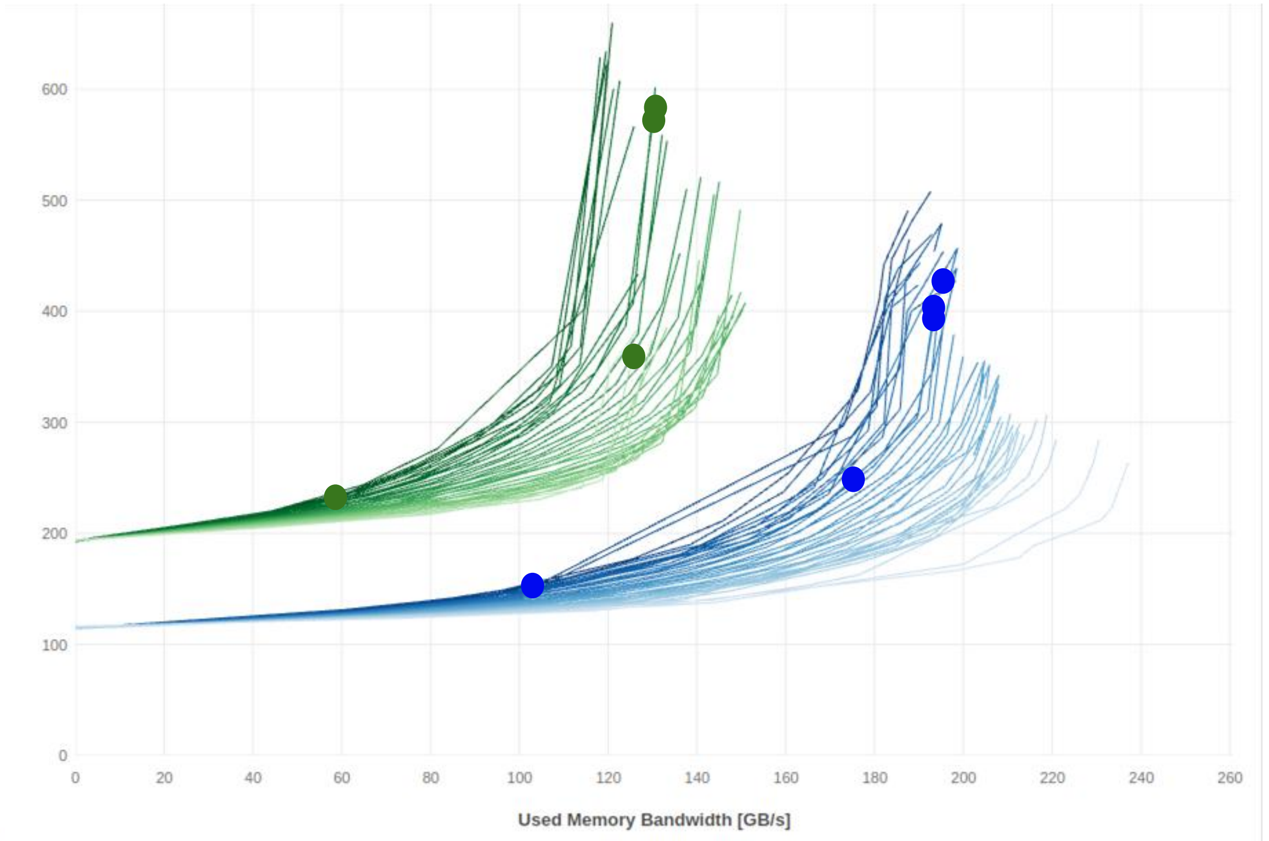




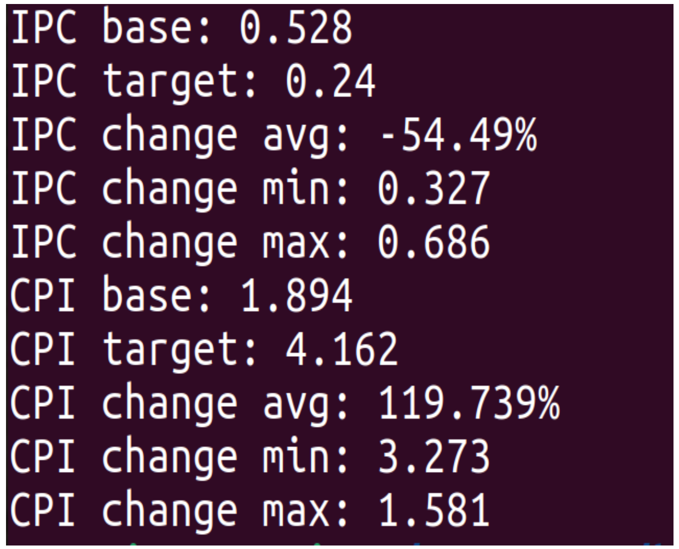


Point by point → Whole application

- **Visual view → Detailed & Intuitive**
  - Not available in current hardware simulators



- **Summary view**
  - Standard performance metrics:
    - IPC/CPI, exe time, speedup
    - Memory bandwidth, latency
    - …



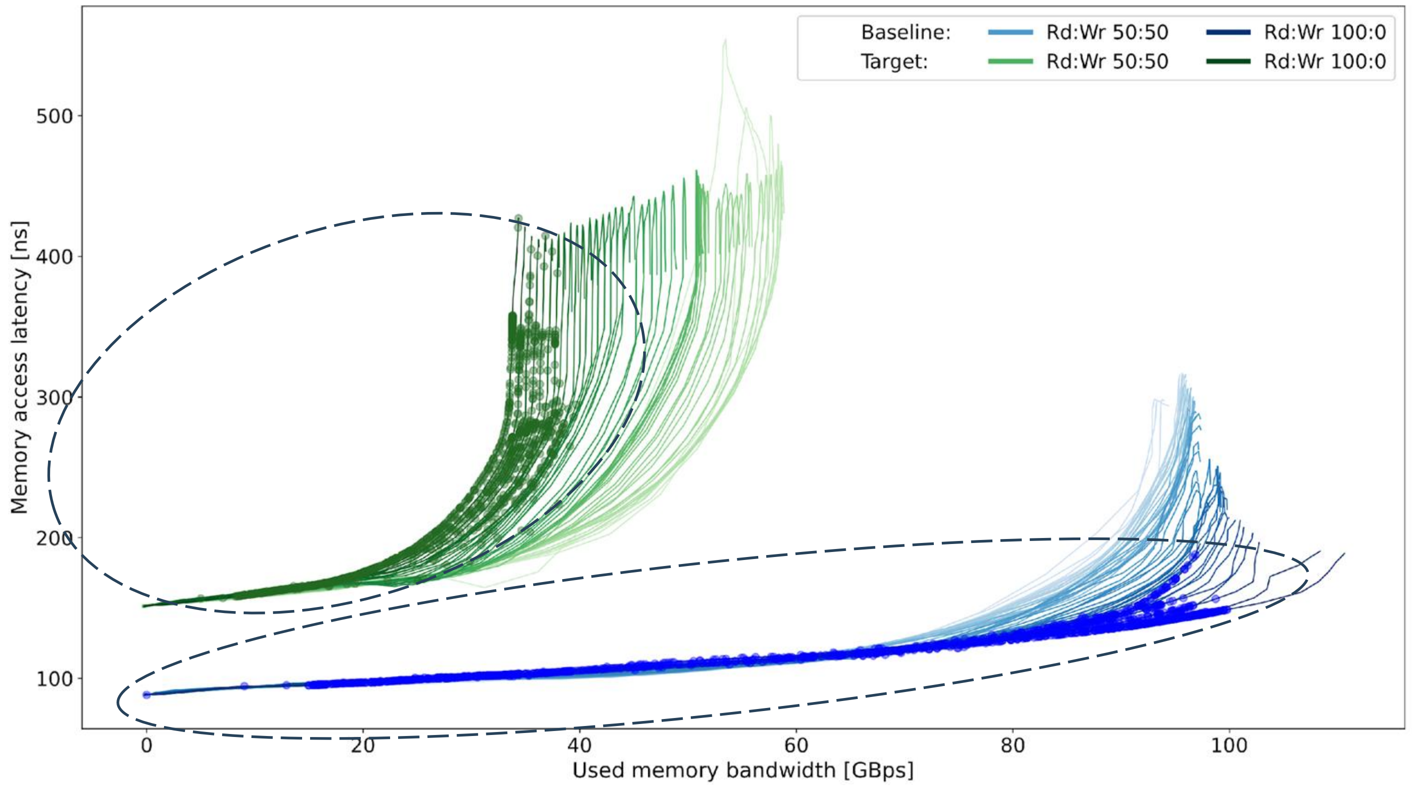
```
IPC base: 0.528
IPC target: 0.24
IPC change avg: -54.49%
IPC change min: 0.327
IPC change max: 0.686
CPI base: 1.894
CPI target: 4.162
CPI change avg: 119.739%
CPI change min: 3.273
CPI change max: 1.581
```

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

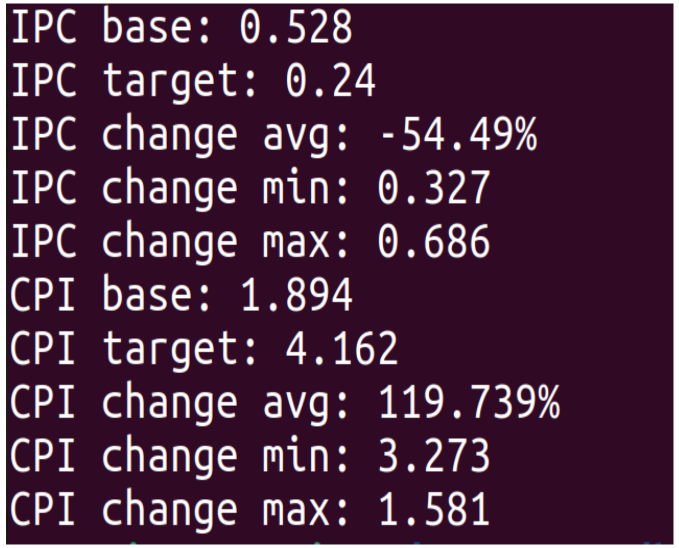
It's the Memory, Stupid!

Point by point → Whole application

- **Visual view → Detailed & Intuitive**
  - Not available in current hardware simulators



- **Summary view**
  - Standard performance metrics:
    - IPC/CPI, exe time, speedup
    - Memory bandwidth, latency
    - …

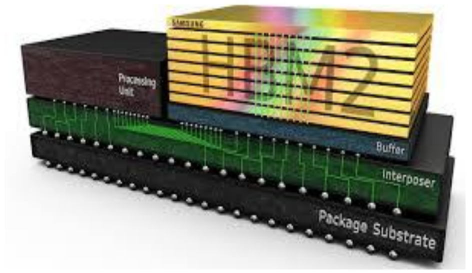


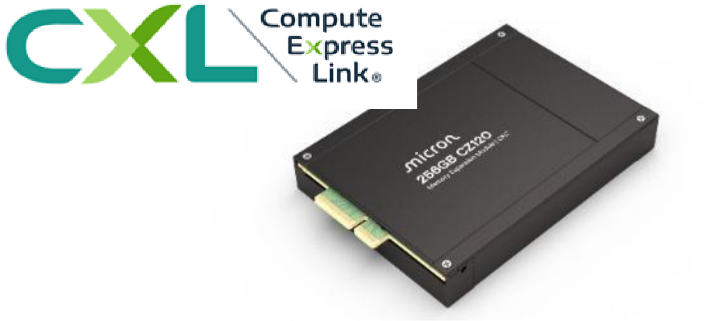
```
IPC base: 0.528
IPC target: 0.24
IPC change avg: -54.49%
IPC change min: 0.327
IPC change max: 0.686
CPI base: 1.894
CPI target: 4.162
CPI change avg: 119.739%
CPI change min: 3.273
CPI change max: 1.581
```

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

It's the Memory, Stupid!

Brutal evaluation





- **Evaluated**
  - **Intel Sandy Bridge** *(DDR3:800/1066/1333/1600)*: Error 5%
  - **Intel Knights Landing** *(DDR4-2400/MCDRAM)*: Error 4%
  - **Huawei Kunpeng 920** *(DDR4 1600/1866/2933)*: Error 4.5%
  - **Intel Cascade lake**
    - *DDR4 alone → DDR4+Optane:* Error 0.7%
    - DDR4 2666 → 2133: Error 0.7%
    - DDR4 → Optane: Error 26%
  - **Intel Emerald Rapids** *(DDR5 4800 → 3200)*: Error 3%
  - **Intel Granite Rapids** *(RDIMM-6400 → MRDIMM-8800)*: Error 4%
- **Work in progress**
  - **Intel Max:** *DDR5-4800 + HBM2*
  - **Intel Emerald Rapids:** *DDR5-4800 + CXL Memory Expander*

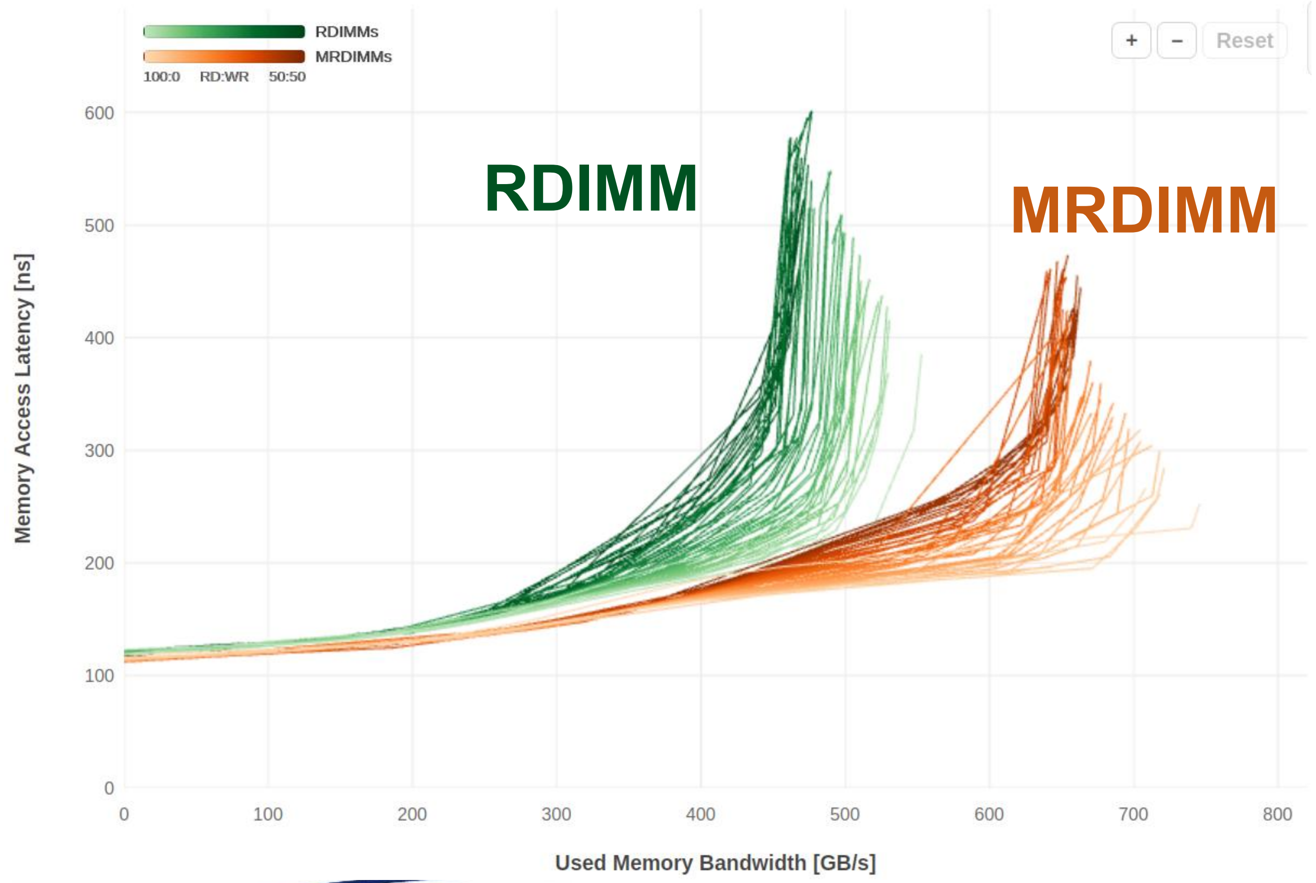
Benchmarks: SPEC, HPC apps, memory intensive benchmarks (STREAM, Google Multichase, etc.)

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

It's the Memory, Stupid!

Intel GNR RDIMM-6400 → MRDIMM-8800

- **Unprecedented memory system performance**
  - *+75% bandwidth utilization: Sustained/Max theoretical → Comparable to or better than servers with DDR4/5 RDIMMs*
  - *Similar latency to DDR4/5 RDIMMs*
  - **Plug&Play**

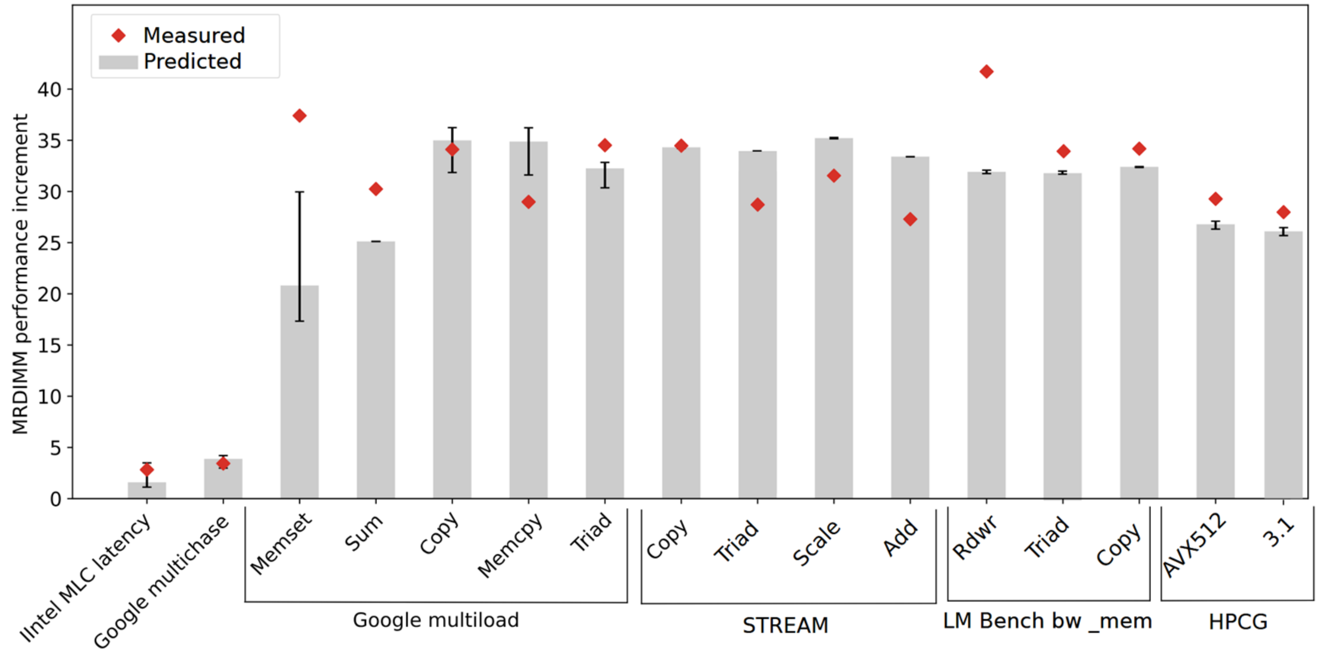


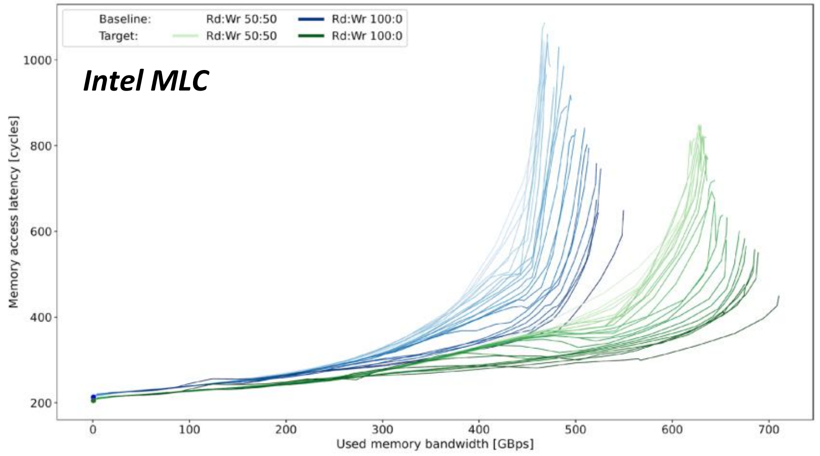
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

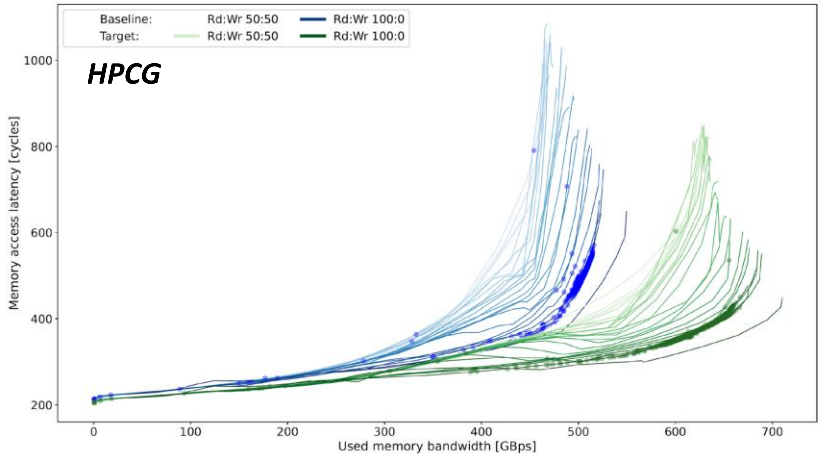
Intel GNR RDIMM-6400 → MRDIMM-8800

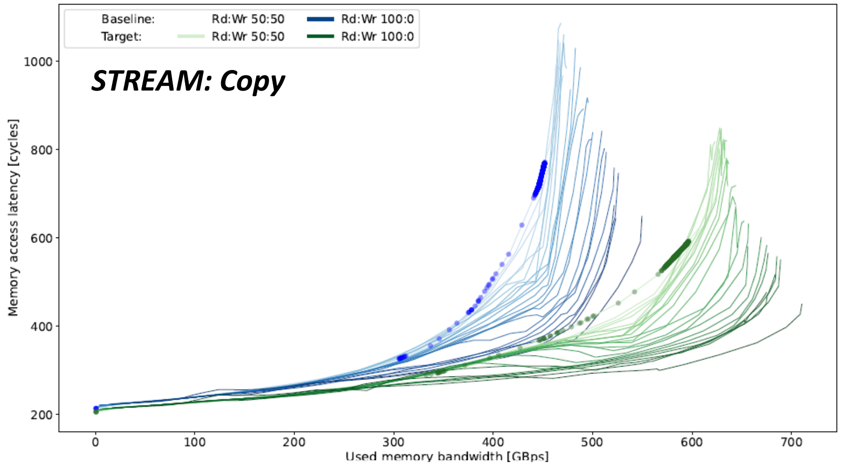
- **Bar chart (right)**
  - Estimation error bars:
    Based on boundaries for MLP and OOO-execution
    (we cannot measure the exact values)

- **Application segments (examples)**

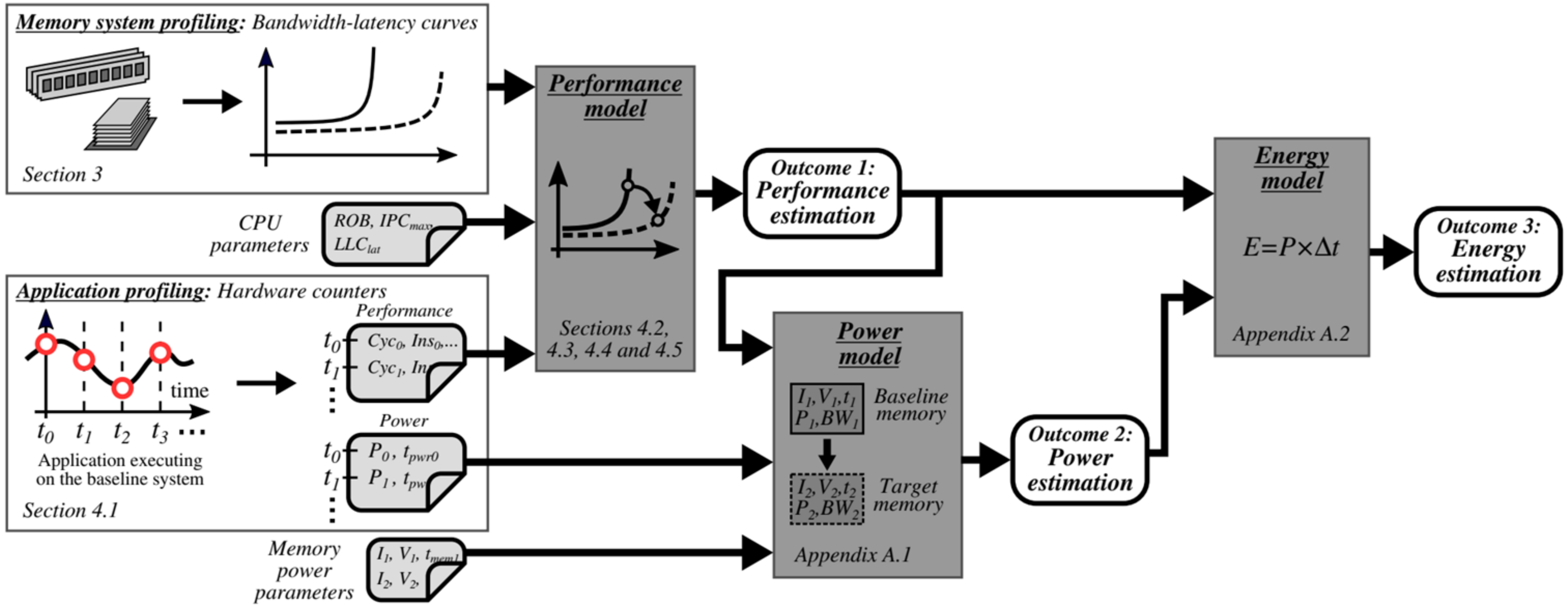








PROFET predicts: Performance, power and energy


Memory system profiling: Bandwidth-latency curves
Section 3
CPU parameters
ROB, IPC$_{max}$, LLC$_{lat}$
Application profiling: Hardware counters
time
t_0 t_1 t_2 t_3 ...
Application executing on the baseline system
Section 4.1
Performance
t_0 Cyc$_0$, Ins$_0$,...
t_1 Cyc$_1$, In
Power
t_0 P_0, t_{pwr0}
t_1 P_1, t_{pw}
Memory power parameters
I_1, V_1, t_{mem1}
I_2, V_2
Performance model
Sections 4.2, 4.3, 4.4 and 4.5
Outcome 1: Performance estimation
Power model
I_1,V_1,t_1 P_1,BW_1 Baseline memory
I_2,V_2,t_2 P_2,BW_2 Target memory
Appendix A.1
Outcome 2: Power estimation
Energy model
$E=P\times\Delta t$
Appendix A.2
Outcome 3: Energy estimation

- This presentation focused on **performance**
- **Power energy**: paper, git, follow-up presentation

Why PROFET is cool

- **Disruptive approach to memory design space exploration** (hardware simulation)

- Instead of simulating your application on a baseline hardware → **Run & Profile**
  - Simpler, faster, scalable, more correct

- **Memory bandwidth-latency curves**
  - Not trivial (especially in 2019)
  - But still, much simpler, faster, scalable and more correct than memory systems simulation

- Designed for SandyBridge with DDR3 → **Plug&Play "everywhere"**
  - See the evaluation slide → The PROFET did not change from SandyBridge
  - The PROFET model is
    - Generic: Requires standard CPU parameters (datasheets) and profiling (hardware counters)
    - Formal: No "magic numbers" which vary from one system to another

- **Brutally evaluated against actual hardware**

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

Questions?

mariana.carmin@bsc.es

Demo / Hands-on

Follow along!

rb.gy/s6hfzr

- We are going to be showing the following apps
  - STREAM Triad
    - https://www.cs.virginia.edu/stream/FTP/Code/
  - Google Multichase
    - https://github.com/google/multichase
- On the following Systems
  - **MN5-GPP** -
    - Sapphire rapids w/ 8x **16GB** DDR5-4800
  - **MN5 GPP-HighMem**
    - Sapphire rapids w/ 8x **64GB** DDR5-4800
  - **Intel Granite Rapids** -
    - Granite rapids w/ 12× **64GB RDIMM**–6400
  - **Intel Granite Rapids**
    - Granite rapids w/ 12× **64GB MRDIMM**–8800

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

It's the Memory, Stupid!

Demo / Hands-on

Follow along!

rb.gy/s6hfzr

- **Demo 1**:
  - MN5-GPP → MN5 GPP-HighMem
- **Demo 2:**
  - RDIMM → MRDIMM

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

It's the Memory, Stupid!

Demo / Hands-on

Follow along!

rb.gy/s6hfzr

Let’s see it…

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

It's the Memory, Stupid!

Explore more

Pointers to related work

- PROFET paper: Milan Radulovic, Rommel Sánchez Verdejo, Paul Carpenter, Petar Radojković, Bruce Jacob, and Eduard Ayguadé. ***PROFET: Modeling System Performance and Energy Without Simulating the CPU***. SIGMETRICS 2019.
- PROFET github: https://github.com/bsc-mem/PROFET
- PROFET sensitivity analysis: https://esecretary.salle.url.edu/zona_autentica/memoriesTFC/25299_GI_2023.pdf

Material page: memory.bsc.es/itms/materials

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

Questions?

mariana.carmin@bsc.es

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

Thank you!

mariana.carmin@bsc.es